REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-6 are pending in the application. The Examiner additionally stated that claims 1-6 are rejected. By this amendment, claims 1-6 have been amended and claims 7-23 have been added. Hence, claims 1-23 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments and the following arguments.

In the Specification

The Examiner objected to the specification indicating that Applicant should update the status of the application on page 1 of the specification, as appropriate. In that the status of the application has not changed, no update is required. Applicant therefore requests that the objection to the specification be withdrawn.

In addition, Applicant has amended the specification to secure a substantial correspondence between the claims amended herein and the remainder of the specification. No new matter is presented.

In the Claims

Rejections Under 35 U.S.C. §102(b)

The Examiner rejected claims 1-6 under 35 U.S.C. 102(b) as being anticipated by Quattromani et al. (5,471,598), hereinafter, "Quattromani." Applicant respectfully traverses the Examiner's rejections.

With regard to claims 1, 3, and 5, the Examiner stated that Quattromani teaches a unified cache memory 70 which stores data ("data cache") attached to a write buffer ("bypass system"). The Examiner noted that write buffer includes two latches, 162x and 162y, which output values to the cache ("two ports to the data cache"), a plurality of write buffer entries, 152x0-x5 and 152y0-y5, ("registers for multiple data entries"), a bus connecting the write buffer 27 with the core 20, and write buffer control logic 150 ("address matching and switching logic"). The Examiner indicated that each write to the memory or cache (including write hits) are stored in the write buffer 27 ("characterized in

that write operations that hit . . . written to the data cache"). Furthermore, the Examiner state that reads after writes are compared in the write buffer control logic and if a physical address of data stored in a entry of the write buffer matches a physical address of a read operation from the core (see col. 21, lines 17-21 and 32-39), then it is determined if the entry or entries are bypassable (see col. 22, lines 42-60). If the entry is bypassable, then a process is performed by the write buffer control logic 150 to enable the sourcing of contents of the data portion of the write buffer entry directly to the core 20 ("read operations use the address matching logic . . . stored in the data cache memory array"). See col. 22, line 65 - col. 23, line 3 and fig. 11. The Examiner noted that each write of an entry in the write buffer to the data buffer results frees the ports to the used in further write operations.

Amended claim 1 is provided below for ease of reference.

1. A multi-streaming microprocessor core, for executing instruction streams running within the multi-streaming microprocessor core at any time, the multi-streaming microprocessor core comprising:

instruction queues, each corresponding to each of the instruction streams, said each of said instruction queues comprising:

first instructions, for dispatch to one or more functional units;

store instructions, for dispatch to a data cache, wherein said store instructions direct write operations; and

load instructions, for dispatch to said data cache, wherein said load instructions direct read operations;

a bypass structure within said data cache, for receiving said store instructions, said bypass structure comprising multiple elements, wherein, if said write operations hit in said data cache, data corresponding to said write operations are stored in one or more of said elements in said bypass structure before said data is written to said data cache; and

address matching logic, coupled to said bypass structure within said data cache, for receiving said load instructions, wherein said read operations use said address matching logic to search said elements of said bypass structure to identify and use any one or more of said elements representing more recent data than that stored in said data cache.

Applicant respectfully disagrees with the Examiner's rejection of claim 1 for the following reasons. First, Quattromani's teachings are directed toward a superscalear superpipelined microprocessor having a write buffer located between the core and the cache. The write buffer is controlled to store the results of write operations to memory until such time as the cache becomes available, such as when no high-priority reads are to be performed. See Abstract. The microprocessor has a CPU core 20 in a superscalar design with two execution pipes X and Y. It includes an instruction decoder 21. See col. 5, lines 30-31. A pipe control unit 28 controls instruction flow through the execution pipes, including keeping the instruction s in order until it is determined that an instruction will not cause an exception, squashing bubbles in *the instruction stream*. See col. 5, lines 45-48. Quattromani describes an instruction fetch stage that provides a continuous *code stream* into the CPU core. See col. 5, lines 60-61.

Clearly, the microprocessor core taught by Quattromani is directed toward execution of a single instruction stream. See in addition Figs. 1-2, and 4. He describes a write buffer that is located between the core and the cache, whose function is to receive address and data information from the core 20 that are to be written to memory, rather that to one of the registers in register file 24; the address and data information stored in the write buffer can then be later written to memory at such time as the cache and memory subsystems are not otherwise busy in a higher priority operation. See col. 8, lines 60-67.

Applicant respectfully submits that Quattromani does not disclose instruction queues, each corresponding to each of the instruction streams. This is because Quattromani teaches a single instruction stream. In that Quattromani's teachings address execution of a single instruction stream, it follows that he does not describe, within each of the instruction queues: first instructions, for dispatch to one or more functional units; store

instructions, for dispatch to a data cache, where the store instructions direct write operations; and load instructions, for dispatch to the data cache, where the load instructions direct read operations. Furthermore, Quattromani utterly fails to disclose a bypass structure within the data cache, for receiving store instructions from each of the instruction queues. Nor does he allude to address matching logic, for receiving load instructions from each of the instruction queues. Applicant has moreover searched the teachings of Quattromani to discover any motivation, suggestion, or even a hint that would lead one skilled in the art toward providing a bypass structure within a data cache that receives store instructions from more than a single instruction stream. Quattromani is silent on the subject of multi-streaming microprocessor cores and thus does not provide any motivation whatsoever to appreciate the problems addressed by the present invention, that is, how to eliminate stalls in read and write operations to a data cache within a multi-streaming microprocessor core.

For these reasons, Applicant respectfully requests that the Examiner withdraw his rejection to claim 1.

With respect to claim 2, this claim depends from claim 1 and adds further limitations that are neither anticipated nor made obvious by Quattromani. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejection of claim 2. Likewise, added claims 7-13 depend from claim 1 as well and add further limitations that are neither anticipated nor made obvious by Quattromani.

Claim 3 is repeated below for ease of reference.

3. A multi-streaming microprocessor core, for executing instruction streams running within the multi-streaming microprocessor core at any time, the multi-streaming microprocessor core comprising:

instruction queues, each corresponding to each of the instruction streams, said each of said instruction queues comprising:

first instructions, for dispatch to one or more functional units;

store instructions, for dispatch to a data cache, wherein said store instructions direct write operations; and

load instructions, for dispatch to said data cache, wherein said load instructions direct read operations;

a bypass structure within said data cache, for receiving said store instructions, said bypass structure comprising multiple elements, wherein, if said write operations hit in said data cache, data corresponding to said write operations are stored in one or more of said elements in said bypass structure before said data is written to said data cache;

address matching logic, coupled to said bypass structure within said data cache, for receiving said load instructions, wherein said read operations use said address matching logic to search said elements of said bypass structure to identify and use any one or more of said elements representing more recent data than that stored in said data cache; and

switching logic, coupled to said bypass structure within said data cache, for determining where a newest version of said more recent data resides based on bytes, and wherein one of said read operations matches on multiple elements of said bypass structure.

Applicant respectfully disagrees with the Examiner's rejection of claim 3 for reasons substantially similar to those cited above in disputation of the Examiner's rejection of claim 1. In essence, Quattromani discloses and describes execution of a single instruction stream and therefore fails to disclose instruction queues, each corresponding to each of the instruction streams. Nor does he teach, or even suggest, a bypass structure within the data cache, for receiving store instructions from each of the instruction queues. His teachings do not extend to address matching logic, for receiving load instructions from each of the instruction queues either. Furthermore, Quattromani does not describe, nor allude to, switching logic that determines where a newest version of more recent data resides based on bytes, and where one of the read operations matches on multiple

elements of the bypass structure. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejection of claim 3.

With respect to claim 4, this claim depends from claim 3 and adds further limitations that are neither anticipated nor made obvious by Quattromani. Therefore, it is respectfully requested that the Examiner withdraw his rejection of claim 4. Applicant furthermore notes that new claims 14-17 depend from claim 3 and add further limitations that are neither anticipated nor made obvious by Quattromani.

Amended claim 5 is provided below for ease of reference.

5. A method for eliminating stalls in read and write operations to a data cache within a multi-streaming microprocessor core, comprising:

providing multiple instruction streams to corresponding instruction queues, said providing comprising:

first dispatching first instructions to one or more functional units;

second dispatching store instructions to a data cache, wherein the store instructions direct write operations; and

third dispatching load instructions to the data cache, wherein the load instructions direct read operations;

first receiving the store instructions in a bypass structure within the data cache, wherein the bypass structure comprises multiple elements, and wherein, if the write operations hit in the data cache, storing data corresponding to the write operations in one or more of the elements in the bypass structure before the data is written to the data cache; and

second receiving the load instructions in address matching logic within the data cache, wherein the read operations use the address matching logic to search the elements of the bypass structure to identify and use any one or more of the elements representing more recent data than that stored in the data cache.

Applicant respectfully disagrees with the Examiner's rejection of claim 5 for reasons substantially similar to those cited above in disputation of the Examiner's rejection of claims 1 and 3. Because Quattromani discloses and describes execution of a single instruction stream via a single instruction queue, he does not teach providing multiple instruction streams to corresponding instruction queues. Nor does he disclose dispatching load and store instructions from multiple instruction queues to a data cache. Likewise, Quattromoni fails to teach receiving the store instructions from the multiple instruction queues in a bypass structure within the data cache or receiving the load instructions from the multiple instruction queues in address matching logic. In that Quattromani is silent on the subject of a microprocessor that is capable of executing multiple instruction streams, Applicant respectfully requests that the rejection of claim 5 be withdrawn.

With respect to claim 6, this claim depends from claim 5 and adds further limitations that are neither anticipated nor made obvious by Quattromani. Therefore, it is respectfully requested that the Examiner withdraw his rejection of claim 6. Applicant furthermore notes that new claims 18-23 depend from claim 5 and add further limitations that are neither anticipated nor made obvious by Quattromani.

CONCLUSIONS

In view of the arguments advanced above, Applicant respectfully submits that claims 1-23 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

EXPRESS MAIL LABEL NUMBER: **EO 002 322 365 US**
DATE OF DEPOSIT: **7/27/2004**
I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to Mail Stop PETITION, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Respectfully submitted,

HUFFMAN PATENT GROUP, LLC

By _____
RICHARD K. HUFFMAN
Reg. No. 41,082
Tel.: (719) 575-9998

Date: 7/27/04